Evaluator Performance

Tracking Evaluator Performance

- So far running the Evaluator took very long
  - Central events could only be run with restricting the matching to embedded tracks only.
    - Ok, for efficiency studies
    - Impossible to evaluate fake rates
    - CPU need for full evaluation
      - in SEMI central events
        - Tracking takes ~180 sec
        - Evaluator took ~2700sec (full event)
      - In central events
        - Tracking ~5min, 11 min including clustering
        - Evaluator > 10h
- Note with the current TPC cell sizes we can't run the simulation and write the clusters to disk.
  - Job blows up to 20GB and hangs
  - 1 event probably too big to write to a single file
  - This makes working on the tracking in central events very painful
- Can we put a timing service in Fun4AllServer?

Tracking Evaluator Options

- Matches reco and truth information and dumps information in Ntuples (defaults in Setup macro):
  - _ntp_vertex; (true)
  - _ntp_gpoint; (false by default)
  - _ntp_g4hit; (true)
  - _ntp_hit; (true)
  - _ntp_cluster; (true)
  - _ntp_gtrack; (true)
  - _ntp_track; (true)
- Options:
  - _scan_for_embedded,
    - Scan for embed (true) -> Only look at embedded particles
  - _do_track_match
    - Allows to turn off matching to only dump reco objects into ntuples (fast!)

- Check again if the default settings are what they should be
    - Skip items that are rarely use to save CPU for most users

Optimization done so far

Using semi central events as benchmark

- _ntp_vertex; -> fast, 0.26sec, nothing done
- _ntp_cluster; -> was 666sec, now is 1.3sec
  - Ineffcient loop in trackeval->best_track_from(cluster)
    - Was: For each cluster loop over all tracks and all clusters to find which track best matches the cluster and cache the matches
    - Now is: Loop over all tracks once and cache the matches
  - Turned off (by default) the lookup of the last hit of the matching geant track.
    - Needs lookup of geant track and then loop over geant hits to find the outermost hit. Even when caching the track match and last hit per geant track this still takes 120sec per event.
    - Added switch -> _do_eval_light(true) turns off this calculation to save time
- _ntp_g4hit; -> was 866sec, now is 8.6sec (1.2 with smaller z matching range, currently using 5cm)
  - Was: loop over all clusters to find the matching one.
  - Most time spent in if(cluster->layer()!= g4hit_layer) continue
  - Now is: Presorted clustermap in layer and z to restrict number of candidates for matching
  - NOTE: getting entries out of the clustermap is an expensive operation. This must change.
- _ntp_gtrack and _ntp_track -> was ~1300sec now is: ~145sec
  - execute faster due to faster g4hit/cluster matching
- _ntp_gpoint and _ntp_hit not touched yet

- Total CPU time for default Evaluator Options (scan for embed -> false)
  - Was: 2700 sec Now is 465 sec
  - 310 sec spent on _ntp_hit -> still needs optimizing or could even be disabled…

- Central events: (scan for embed -> false): 3900sec (with hit match) 1200sec (w/o hit match)
  - Should be ok to run in batch jobs at least for now
  - Still significant room to improve